What if Log4Shell were to happen today?

Piotr Karwasz, VP Logging, Apache Software Foundation: pkarwasz@apache.org
Piotr Karwasz, freelancer: piotr@copernik.eu

Who are we?

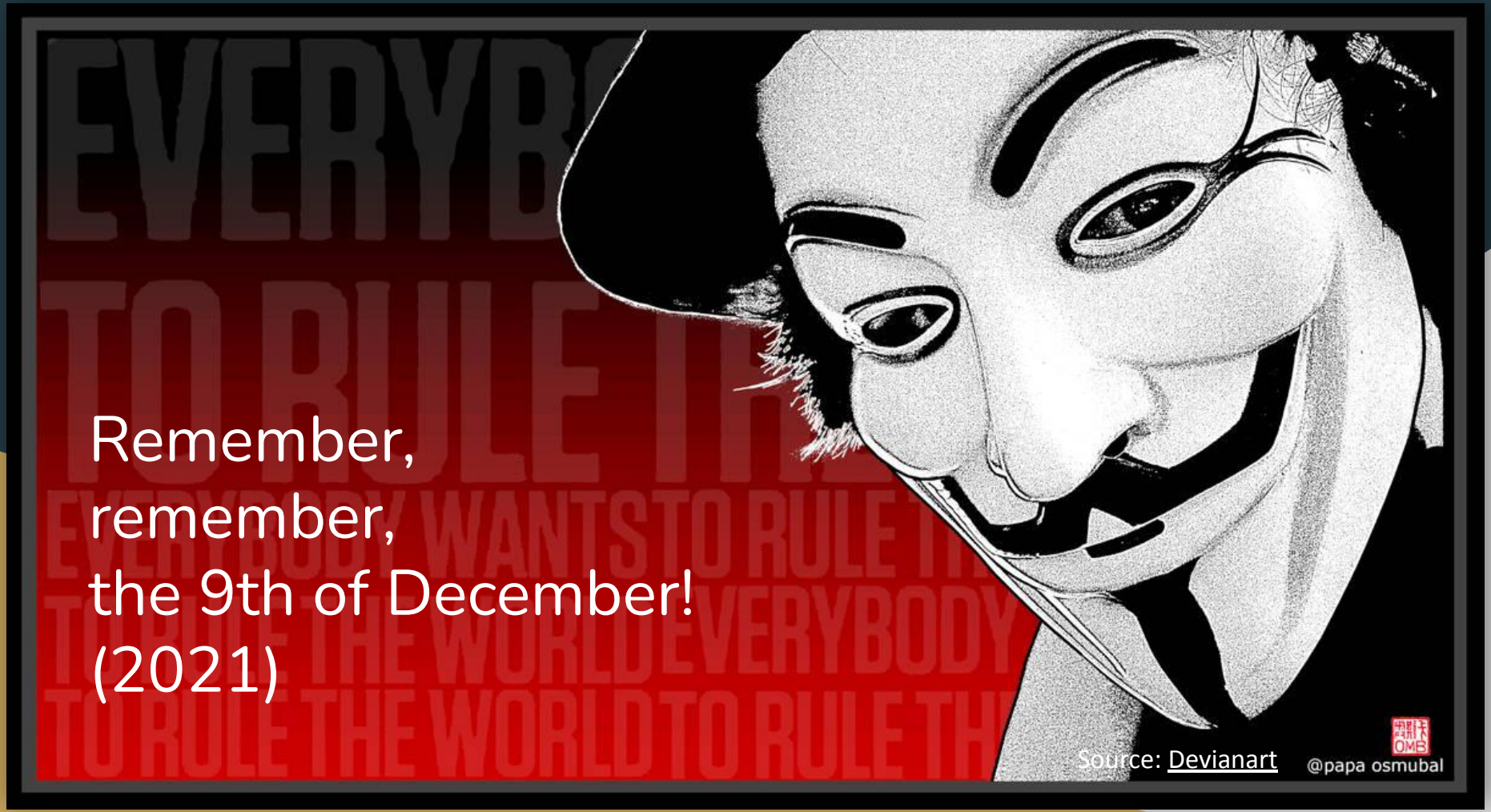
- One of the logging libraries of Apache Logging Services, together with Log4cxx, Log4Net, Log4j Kotlin, Log4j Scala.
- 2001: Ceki Gülcü creates Log4j 1
- 2005-2011: Ceki Gülcü starts working on SLF4J/Logback successor
- 2014: Log4j 2 API/Core is published by: G. Gregory, R. Goers, R. Popma, M. Sicker and others
- 2015: end-of-life of Log4j 1

https://logging.apache.org/log4j/2.x/index.html

Piotr Karwasz:

- 2000: OSS *aficionado*.
- 2009: Ph.D. in Mathematics (UHP, Nancy).
- Father of three daughters: Mimi, Lili and Nati.
- 2017: I started my own IT company.
- 2022, January: start contributing to Log4j.
- 2022, July: Logging Services PMC member.
- 2024, March: ASF member.
- 2024, June: Logging Services PMC chair.

https://oss.copernik.eu/
https://linkedin.com/in/ppkarwasz/


Remember,
remember,
the 9th of December!
(2021)
Source: Devianart
@papa osmubal

CVE-2021-44228 (Log4Shell)

"An attacker who can control log messages or log message parameters can execute arbitrary code loaded from LDAP servers…"—NVD Database

Ingredients:

1. (Pluggable) lookups: `${sys:user.name}`, `${jndi:java:comp/env/value}`
2. (Pluggable) message patterns: `%m` (prints log message), `%d` (prints date), `%p` (prints log level), etc.

Order of evaluation was inverted:

1. Configured pattern:
   `%d ${sys:user.name}: %m`
2. Message pattern evaluated:
   `2024-02-01 ${sys:user.name}: Hello FOSDEM!`
3. Lookups evaluated:
   `2024-02-01 piotr: Hello FOSDEM!`

This flow was reported in LOG4J2-905 (November 2014), classified as feature and a new configuration option was added to disable it.

Timeline of 2.15.0 release

November 24th, 7:51 UTC:
Chen Zhaojun reports the vulnerability

November 24th, 17:30 UTC:
Team discusses the report. It is bad.

November 25th: Thanksgiving!

November 26th, 4:00 UTC:
CVE number requested.

November 30th:
Patch supplied (public PR).

December 5th:
Patch amended, reviewed and merged.

December 7th:
Release vote for `2.15.0` RC1 (72 hours)

December 9th:
Users notice the PR solves a security issue.
Problem with RC1, RC2 vote (7 hours)
Version `2.15.0` released with 7 votes.

Note: Release `2.15.0` was the first of 4 releases that patched a total of 4 CVEs and ended on December 28th with the `2.17.1` release.

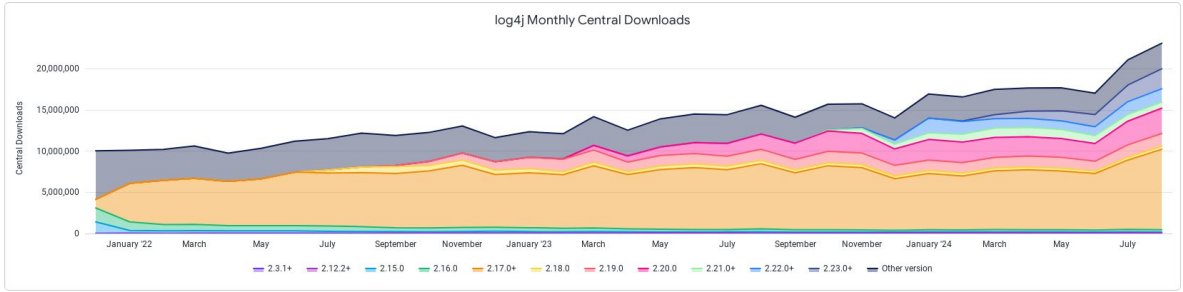
log4j Latest Statistics

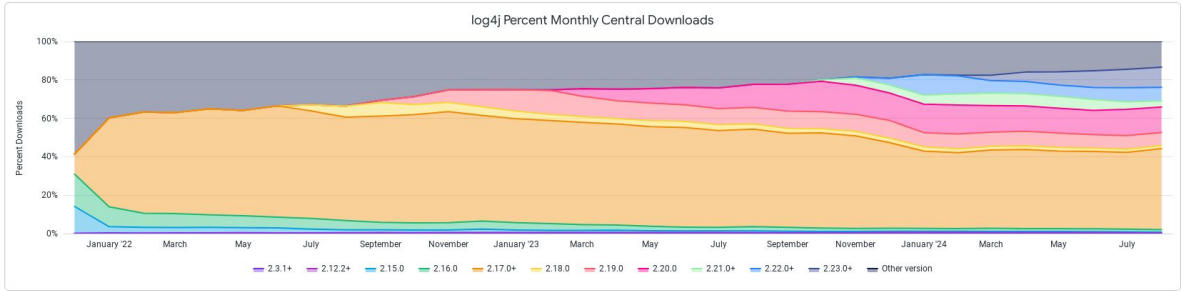
471,179,072

Total Downloads Since Dec 10, 2021
24 % vulnerable

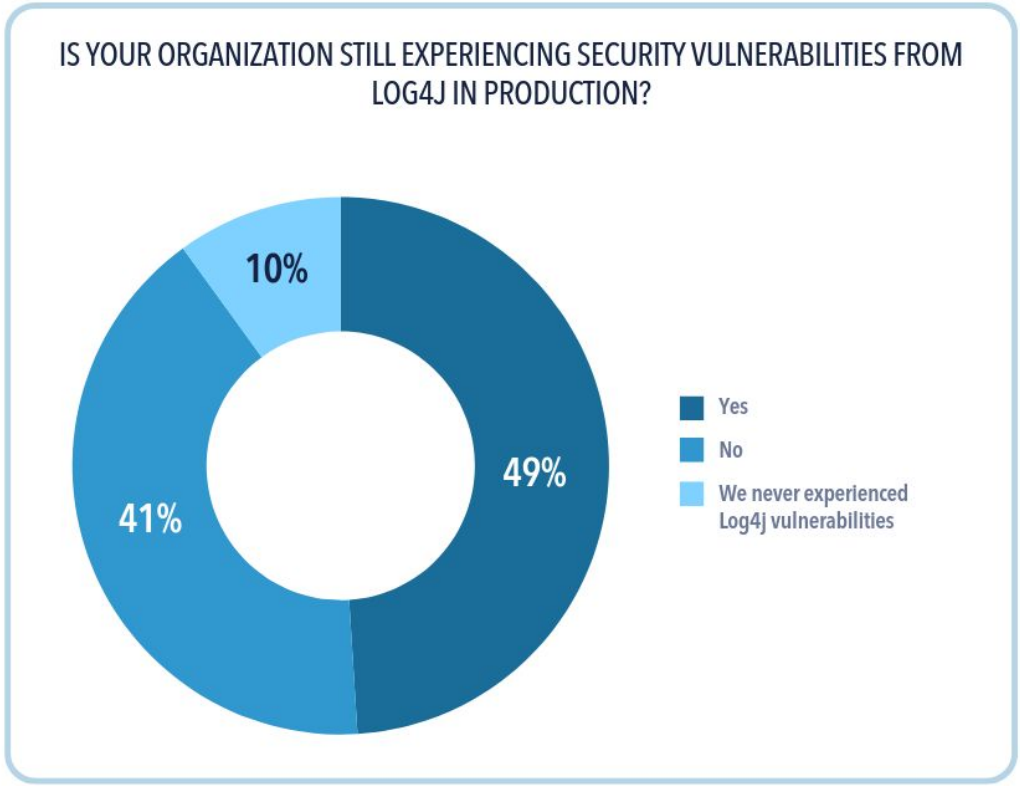
16 %

Vulnerable Downloads Last 7 Days
5,239,729 total downloads





Source: Sonatype [Log4j Updates and Vulnerabilities]



Source: Azul State of Java 2025 Report

Timeline summary

Apache Logging PMC:

- 15 days from report to release
- 11 days to create/merge patch (-4 days for Thanksgiving?)
- 9 days of public patch exposure
- 2 days to prepare a release candidate
- 72 hours for the voting procedure

Users:

- 50% of users downloaded a **patched** version in 30 days (probably 3 times)
- 20% of users downloaded a **vulnerable** version 3 years after the CVE.

Apache Log4j Reactions

How can we do better?

Lessons learned

Supply chain problems:

- Tests are flaky (slow down release),
- Site generation is slow,
- Release procedure is complex,
- Keep dependencies up-to-date (and tell about it).

Too many bundled features:

learn to say NO (intelligently).

Documentation problems:

- Is hard to find,
- Is not complete, some obscure features are not documented,
- Does not contain best practices.

Helped solving the problems:

- **Tidelift** supports Log4j since January 2023,
- German **Sovereign Tech Agency** with a grant to Christian Grobmeier, Volkan Yazıcı and me, since September 2023.

Making a release

Before, a Release Manager had to:

- Select the changes for a new release,
- Run all the test suites,
- Build the website,
- Sign the release,
- Prepare the release notes,
- Handle the voting procedure,
- Release the new version.

Now:

- Select the changes for a new release,
- Prepare the release notes,
- Handle the voting procedure,
- Release the new version.

Future: **Apache Trusted Releases Platform** will also handle voting and releasing the artifacts for us.

Key release elements

- Can we trust automation?
  ASF policy requires the RM to create the binaries.
  Reproducible Builds Project:
  All our Java builds are reproducible!
- Dependabot: upgrades dependencies since 2017.
  We accept those upgrades automatically if tests pass.
- GitHub Actions is the CI/CD engine we use.
- Lots of Maven plugins and test libraries
  that don’t get credit enough!

Testing suite

In December 2021 site generation took hours (rebuild for each Maven Site module).

September 2023:

- Sequential tests,
- Only **unit/integration** tests,
- 30-40% of test runs failed for no reason,
- Build times up to 60 minutes.

September 2024:

- Parallel tests,
- **Dynamic tests (fuzzing)**,
- 8% of test runs fails (21% flaky),
- Build + deploy around 30 minutes.
- Searchable build failure database: Gradle Develocity

Securing optional features

Handling features is hard:

- Features bring users,
- Features bring security exposure,
- OSS is a meritocracy:

  Maintainers have the right to their features in exchange for their work.

- Log4j created a `3.x` branch in 2018 to split each optional dependency, including JNDI into its own artifact.
  Completed: IX 2024
- Removal of seldom downloaded artifacts.
- Ramp-up program:

  We accept new modules with a proven user base and a maintainer. These modules start as third-party.

Vulnerability reporting

Software Bill of Materials

For a library that does **not include** its dependencies in the published JAR, an SBOM has a very limited usage for **vulnerability handling**.

Present:

- Publishing of SBOMs for all Log4j artifacts. The dependency versions are just a **suggestion.**
- Usage of SBOM links to point to a machine-readable VDR.
- Features contributed back to CycloneDX Maven Plugin 2.8.0

Near future:

- Compare information in our SBOM with SBOMs of our dependencies.
- Enrich information in our SBOM with information from dependency SBOMs.
- Download all VDR/VEX metadata from dependency SBOMs.

Work with Christian Grobmeier:
https://github/sbom-enforcer/sbom-enforcer

SBOMs future (?)

- Integration of SBOMs into ecosystem-specific dependency management systems.

Transparency Exchange API for:

- Automatically import VDR/VEX entries from dependencies to stage VEX entries. “Vulnerability Bot”
- Push our VDR, VEX and version suggestions to consumers/dependents.

Security through education

Logging is not always safe:

- Unstructured logging:
  CWE-93 CRLF Injection
- Presence of sensitive information in logs:
  CWE-215: SI in Debug Code
- Injection of {} Log4j formatting patterns:

```
String user = “root {}”;
String what = “login”;
log.(user + “failed to {}”, what);
```

- Reliable and secure transport.

Solutions:

- Rewrite of documentation website.
  Learn from the source, not ChatGPT.
- Generation of reference from code:
  Living documentation,
  Developers can not forget.
- Provide best practices and tips:
  The maintainers knowledge base was mainly unwritten.

Tip: there will be an in-depth book by Christian Grobmeier published by Manning.

Future timeline

Day 0:

- Request a CVE number.
- Start a 72 hours consensus gathering period for shorter vote.
- Establish **private** Git repo (INFRA).

Day 1:

- Propose “fallback” patch (that removes the functionality).

Day 2:

- Optionally propose a better patch.

Day 5-6:

- Accept “fallback” patch if there is not consensus on a better alternative.
- Prepare a release candidate

Day 6-7:

- Log4j **consumers** automatically test the release candidate.

Day 7-9:

- Release and CVE announcement.

Future timeline summary

Apache Logging PMC:

- ~~15~~ **7** days from report to release
- ~~11~~ **3** days to create/merge patch
- ~~9~~ **1** days of public patch exposure
- ~~2 days~~ **1 hours** to prepare a release candidate
- ~~72~~ **24** hours for the voting procedure

Users:

- 20% of users downloaded a patched version **before** the end of the vote.
- 50% of users downloaded a patched version in 30 days

Q & A

https://logging.apache.org/

Thanks

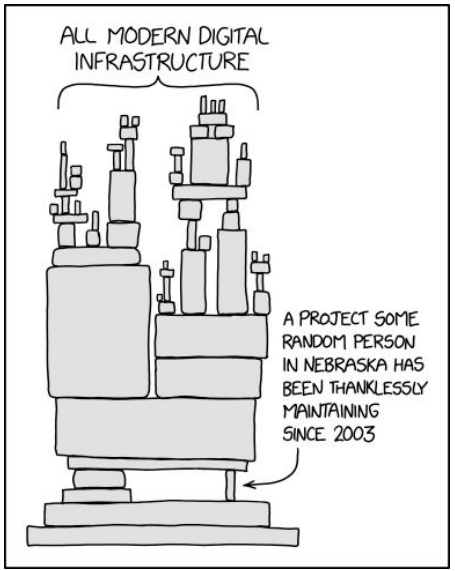
My wife Agnieszka and my angels:
Milena, Liliana, Natalia

Apache Logging Services team:
C. Kozak, D. McColl, D. Psenner, G. Gregory,
J. Friedrich, J. Katariya, M. Sicker, R. Goers, R. Gupta,
R. Popma, R. Middleton, R. Grabowski, S. Deboy,
S. Webb and Th. Schöning.
See also https://logging.apache.org

Partners in crime (STF project):
Christian Grobmeier and Volkan Yazıcı

Financial supporters:
Tidelift and Sovereign Tech Fund

Remember about:



Source: XKCD