<https://github.com/mergebase/log4j-detector>

let op: downloaden van zip-file lukte vanaf mijn laptop niet, wel vanaf kantoor-werkplek Apollo !

Afbeelding met tekst, schermafbeelding, computer

Automatisch gegenereerde beschrijving

<https://github.com/mergebase/log4j-detector.git>

# [mergebase logo](https://raw.githubusercontent.com/mergebase/log4j-detector/master/images/mergebase-small.png)

# Log4-detector

Scanner that detects vulnerable Log4J versions to help teams assess their exposure to [CVE-2021-44228](https://mergebase.com/vulnerability/CVE-2021-44228/), [CVE-2021-45046](https://mergebase.com/vulnerability/CVE-2021-45046/) and [CVE-2021-45105](https://mergebase.com/vulnerability/CVE-2021-45105/). Can search for Log4J instances by carefully examining the complete file-system, including all installed applications. It is able to find Log4J instances that are hidden several layers deep. Works on Linux, Windows, and Mac, and everywhere else Java runs, too!

# Table of Contents

* [Introduction](https://github.com/mergebase/log4j-detector#itemdetector)
* [Example Usage](https://github.com/mergebase/log4j-detector#itemexample)
* [More Example Usage](https://github.com/mergebase/log4j-detector#itemmore)
* [Understanding The Results](https://github.com/mergebase/log4j-detector#itemresults)
* [Usage](https://github.com/mergebase/log4j-detector#itemusage)
* [Build From Source](https://github.com/mergebase/log4j-detector#itembuild)
* [Testing](https://github.com/mergebase/log4j-detector#itemtesting)
* [License](https://github.com/mergebase/log4j-detector#itemlicense)
* [Frequently Asked Questions](https://github.com/mergebase/log4j-detector#faq)
  + [How Does It Work?](https://github.com/mergebase/log4j-detector#itemwork)
  + [This Scanner Only Reports Hits Against The log4j-core Library. What About log4j-api?](https://github.com/mergebase/log4j-detector#itemapi)
  + [Why Report About 2.10.0, 2.15.0, and 2.16.0 ?](https://github.com/mergebase/log4j-detector#item2.10.0)
  + [What are those "file1.war!/path/to/file2.zip!/path/to/file3.jar!/path/to/log4j.jar" results about?](https://github.com/mergebase/log4j-detector#itemwar)
  + [What About Log4J 1.2.x ?](https://github.com/mergebase/log4j-detector#item1.2.x)
  + [How Can I Be Sure This Isn't A Trojan Pretending To Be A Log4J Detector?](https://github.com/mergebase/log4j-detector#itemtrojan)
* [What Is MergeBase All About?](https://github.com/mergebase/log4j-detector#item)

# Introduction

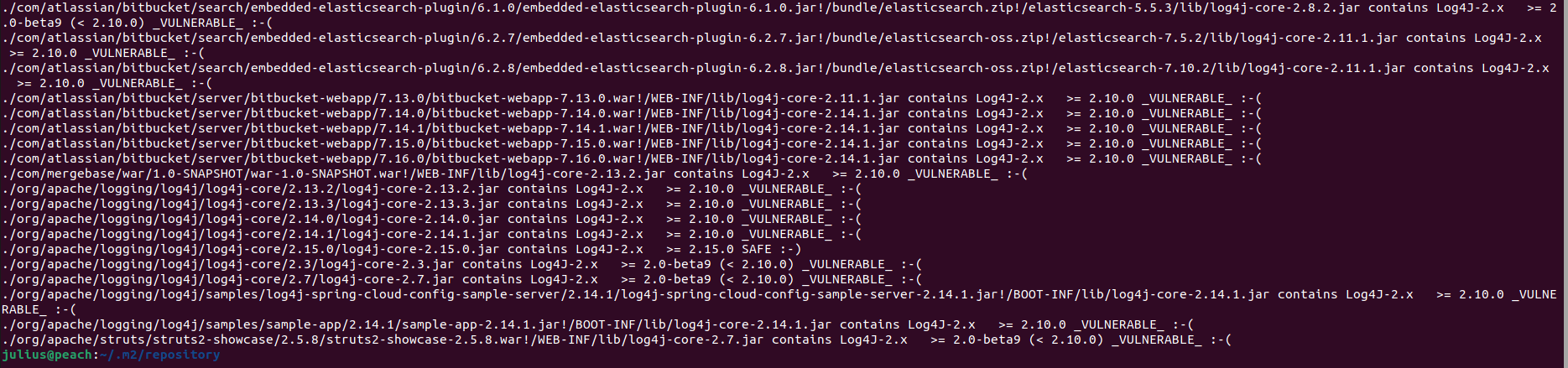
Currently reports log4j-core versions 2.12.2 and 2.17.0 as **\_SAFE\_**, 2.16.0 and 2.15.0 as **\_OKAY\_** and all other versions as **\_VULNERABLE\_** (although it does report pre-2.0-beta9 as "**\_POTENTIALLY\_SAFE\_**").

Can correctly detect log4j inside executable spring-boot jars/wars, dependencies blended into [uber jars](https://mergebase.com/blog/software-composition-analysis-sca-vs-java-uber-jars/), shaded jars, and even exploded jar files just sitting uncompressed on the file-system (aka \*.class).

We currently maintain a collection of [log4j-samples](https://github.com/mergebase/log4j-samples) we use for testing.

# Example Usage:

java -jar log4j-detector-2021.12.20.jar [path-to-scan] > hits.txt

[](https://github.com/mergebase/log4j-detector/blob/master/images/log4j-detector.png)

# More Example Usage:

java -jar log4j-detector-2021.12.20.jar ./samples

-- github.com/mergebase/log4j-detector v2021.12.20 (by mergebase.com) analyzing paths (could take a while).

-- Note: specify the '--verbose' flag to have every file examined printed to STDERR.

/opt/mergebase/log4j-detector/samples/clt-1.0-SNAPSHOT.jar contains Log4J-2.x >= 2.10.0 \_VULNERABLE\_

/opt/mergebase/log4j-detector/samples/infinispan-embedded-query-8.2.12.Final.jar contains Log4J-2.x >= 2.0-beta9 (< 2.10.0) \_VULNERABLE\_

/opt/mergebase/log4j-detector/samples/log4j-1.1.3.jar contains Log4J-1.x <= 1.2.17 \_OLD\_

/opt/mergebase/log4j-detector/samples/log4j-1.2.13.jar contains Log4J-1.x <= 1.2.17 \_OLD\_

/opt/mergebase/log4j-detector/samples/log4j-1.2.17.jar contains Log4J-1.x <= 1.2.17 \_OLD\_

/opt/mergebase/log4j-detector/samples/log4j-core-2.0-beta2.jar contains Log4J-2.x <= 2.0-beta8 \_POTENTIALLY\_SAFE\_ (or did you already remove JndiLookup.class?)

/opt/mergebase/log4j-detector/samples/log4j-core-2.0-beta9.jar contains Log4J-2.x >= 2.0-beta9 (< 2.10.0) \_VULNERABLE\_

/opt/mergebase/log4j-detector/samples/log4j-core-2.0.2.jar contains Log4J-2.x >= 2.0-beta9 (< 2.10.0) \_VULNERABLE\_

/opt/mergebase/log4j-detector/samples/log4j-core-2.0.jar contains Log4J-2.x >= 2.0-beta9 (< 2.10.0) \_VULNERABLE\_

/opt/mergebase/log4j-detector/samples/log4j-core-2.10.0.jar contains Log4J-2.x >= 2.10.0 \_VULNERABLE\_

/opt/mergebase/log4j-detector/samples/log4j-core-2.12.2.jar contains Log4J-2.x >= 2.12.2 \_SAFE\_

/opt/mergebase/log4j-detector/samples/log4j-core-2.14.1.jar contains Log4J-2.x >= 2.10.0 \_VULNERABLE\_

/opt/mergebase/log4j-detector/samples/log4j-core-2.15.0.jar contains Log4J-2.x >= 2.15.0 \_OKAY\_

/opt/mergebase/log4j-detector/samples/log4j-core-2.16.0.jar contains Log4J-2.x >= 2.16.0 \_SAFE\_

/opt/mergebase/log4j-detector/samples/log4j-core-2.4.1.jar contains Log4J-2.x >= 2.0-beta9 (< 2.10.0) \_VULNERABLE\_

/opt/mergebase/log4j-detector/samples/log4j-core-2.9.1.jar contains Log4J-2.x >= 2.0-beta9 (< 2.10.0) \_VULNERABLE\_

# Understanding The Results

**\_VULNERABLE\_** -> You need to upgrade or remove this file.

**\_OKAY\_** -> We only report this for Log4J versions 2.15.0 and 2.16.0. We recommend upgrading to 2.17.0.

**\_SAFE\_** -> We currently only report this for Log4J versions 2.17.0 and 2.12.2.

**\_OLD\_** -> You are safe from CVE-2021-44228, but should plan to upgrade because Log4J 1.2.x has been EOL for 7 years and has several known-vulnerabilities.

**\_POTENTIALLY\_SAFE\_** -> The "JndiLookup.class" file is not present, either because your version of Log4J is very old (pre 2.0-beta9), or because someone already removed this file. Make sure it was someone in your team or company that removed "JndiLookup.class" if that's the case, because attackers have been known to remove this file themselves to prevent additional competing attackers from gaining access to compromised systems.

# Usage

java -jar log4j-detector-2021.12.20.jar

Usage: java -jar log4j-detector-2021.12.20.jar [--verbose] [--json] [--stdin] [--exclude=X] [paths to scan...]

--json - Output STDOUT results in JSON. (Errors/warning still emitted to STDERR)

--stdin - Read STDIN for paths to explore (one path per line)

--exclude=X - Where X is a JSON list containing full paths to exclude. Must be valid JSON.

Example: --exclude='["/dev", "/media", "Z:\TEMP"]'

Exit codes: 0 = No vulnerable Log4J versions found.

1 = At least one legacy Log4J 1.x version found.

2 = At least one vulnerable Log4J version found.

About - MergeBase log4j detector (version 2021.12.20)

Docs - https://github.com/mergebase/log4j-detector

(C) Copyright 2021 Mergebase Software Inc. Licensed to you via GPLv3.

# Build From Source:

git clone https://github.com/mergebase/log4j-detector.git

cd log4j-detector/

mvn install

java -jar target/log4j-detector-2021.12.20.jar

# Testing:

We maintain a collection of log4j samples here: <https://github.com/mergebase/log4j-samples>

# License

GPL version 3.0

# Frequently Asked Questions

# How Does It Work?

The Java compiler stores String literals directly in the compiled \*.class files. If log4j-detector detects a file named "JndiManager.class" on your file-system, it then examines that file for this String: "Invalid JNDI URI - {}". Turns out that specific String literal is only present in the patched version of Log4J (version 2.15.0). Any versions of Log4J without that String are vulnerable.

## This Scanner Only Reports Hits Against The log4j-core Library. What About log4j-api?

Many scanners (including GitHub's own [Dependabot](https://github.com/dependabot)) currently report both "log4j-core" and "log4j-api" libraries as vulnerable. These scanners are incorrect. There is currently no existing version of the "log4j-api" library that can be exploited by any of these vulnerabilities.

At [MergeBase](https://mergebase.com/) we pride ourselves on our scan accuracy. You're already busy enough patching and defending your systems. We don't want you to waste your time with false positives. That's why we don't report any hits against log4j-api.

## Why Report About 2.10.0, 2.12.2, 2.15.0, 2.16.0, and 2.17.0 ?

Version 2.10.0 is important because that's the first version where Log4J's vulnerable "message lookup feature" can be disabled via Log4J configuration.

Version 2.12.2 is important because it's a Java 7 compatible version of Log4J that is not vulnerable to CVE-2021-44228.

Versions 2.15.0 and 2.16.0 are important because these are the first versions where Log4J's default out-of-the-box configuration is not vulnerable to CVE-2021-44228.

And version 2.17.0 is important because it's not vulnerable to more recently discovered CVEs such as CVE-2021-45046 and CVE-2021-45105. Despite these being much less serious vulnerabilities, we anticipate everyone will want to patch to 2.17.0.

## What are those "file1.war!/path/to/file2.zip!/path/to/file3.jar!/path/to/log4j.jar" results about?

The "!" means the log4j-detector entered a zip archive (e.g., \*.zip, \*.ear, \*.war, \*.aar, \*.jar). Since zip files can contain zip files, a single result might contain more than one "!" indicator in its result.

Note: the log4j-detector only recursively enters zip archives. It does not enter tar or gz or bz2, etc. The main reason being that Java systems are often configured to execute jars inside jars, but they are never configured to execute other file formats (that I know of!). And so a log4j copy inside a \*.tar.gz is probably not reachable for a running Java system, and hence, not a vulnerability worth reporting.

2nd note: for zips-inside-zips our scanner does load the inner-zip completely into memory (using ByteArrayInputStream) before attempting to scan it. You might need to give Java some extra memory if you have extremely large inner-zips on your system (e.g., 1 GB or larger).

## What About Log4J 1.2.x ?

Only versions of Log4J 2.x (from 2.0-beta9 to 2.14.1) are vulnerable to CVE-2021-44228.

## How Can I Be Sure This Isn't A Trojan Pretending To Be A Log4J Detector?

Great question! Since we include the complete source code here in Github (all 2200 lines of Java), as well as the steps to build it, and since this tool has zero dependencies, it shouldn't take too long to carefully study the code to your satisfaction. If you don't trust Maven you can go directly into the "src/main/java/com/mergebase/log4j" directory and type "javac \*.java". That works, too!

We also sign the pre-compiled jar we keep in the root of the repository (./log4j-detector-2021.12.20.jar) with the MergeBase code signing key. Please run "jarsigner -verbose -verify log4j-detector-2021.12.20.jar" to confirm this.

# What Is MergeBase All About?

[MergeBase](https://github.com/mergebase/log4j-detector/blob/master/images/mergebase-small.png)

[MergeBase](https://mergebase.com/) is an SCA company (Software Composition Analysis) based in Vancouver, Canada. We're similar to companies like Snyk, Sonatype, Blackduck, etc., in that we help companies detect and manage vulnerable open-source libraries in their software. Check us out! We have great accuracy, great language support, and we're not too expensive, either: [mergebase.com/pricing](https://mergebase.com/pricing/).

We would be delighted if anyone takes a [2-week free trial](https://mergebase.com/try/) of our SCA product! And if you email our CEO ([oscar@mergebase.com](mailto:oscar@mergebase.com)) with the subject "log4j-detector" we will extend your free trial to 4-weeks.

**Controle installatie van JAVA op server:**

C:> Which java

C:> Java -version

After installing JDK on your client, add the directory path to the following environment variables:

* $JAVA\_HOME

This variable must be set to the top directory of the installed JDK base.

* $PATH

This variable must include $JAVA\_HOME/bin.

* $LD\_LIBRARY\_PATH

This variable must include $JAVA\_HOME/lib.

# Let op: git clone commando werkt niet, ik krijg een SSL-fout vanuit git !!!!!!!

# Pak ZIP-file handmatig uit onder GITHUB-WORK/log4j-detector

# Afbeelding met tekst Automatisch gegenereerde beschrijving

# Ga met command-box/git-bash naar directory log4j-detector-master

# Start check:

java -jar log4j-detector-2021.12.20.jar C: >hits-20211221.txt

via git-bash:

java -jar log4j-detector-2021.12.20.jar ./../../../ > hits-20211221.txt

let op: vorige statement loopt stuk omdat hij dan ook naar de ONEDRIVE gaat. Deze directory excluden mbv:

--exclude='["C:\ONEDRIVE-PETER”]'

**Let op: quotjes vanuit WORD werken niet in GITBASH. Als je quotje vanuit GITBASH kopieert naar word, dan gaat het wel !!!!**

./../../../ONEDRIVE-PETER

DIRECTORIES VANAF KANTOOR-PC BROEK:

java -jar log4j-detector-2021.12.20.jar 'c:/Oracle/' >hits-20211221-Oracle.txt

java -jar log4j-detector-2021.12.20.jar 'c:/Oracle64/' >hits-20211221-Oracle64.txt

java -jar log4j-detector-2021.12.20.jar 'c:/oraclexe/' >hits-20211221-oraclexe.txt

java -jar log4j-detector-2021.12.20.jar 'c:/Program Files/' >hits-20211221-programfiles.txt

java -jar log4j-detector-2021.12.20.jar 'c:/Program Files (x86)/' >hits-20211221-programfiles64.txt

java -jar log4j-detector-2021.12.20.jar 'c:/TEMP/' >hits-20211221-temp.txt

java -jar log4j-detector-2021.12.20.jar 'c:/Windows/' >hits-20211221-Windows.txt

java -jar log4j-detector-2021.12.20.jar 'c:/CAD Viewer 11/' >hits-20211221-CADViewer11.txt

java -jar log4j-detector-2021.12.20.jar 'c:/ProgramData/' >hits-20211221-programdata.txt

java -jar log4j-detector-2021.12.20.jar 'c:/ibm/' >hits-20211221-ibm.txt

java -jar log4j-detector-2021.12.20.jar 'c:/install/' >hits-20211221-install.txt

Afbeelding met tekst

Automatisch gegenereerde beschrijving

DIRECTORIES VANAF EIGEN LAPTOP:

Runnen vanuit directory C:\GITHUB-WORK\log4j-detector-master

java -jar log4j-detector-2021.12.20.jar 'c:/' --exclude='["C:/Users","C:/peter"]' >hits-tot-20211223.txt

java -jar log4j-detector-2021.12.20.jar 'c:/Oracle/' >hits-20211221-Oracle.txt

java -jar log4j-detector-2021.12.20.jar 'c:/Oracle64/' >hits-20211221-Oracle64.txt

java -jar log4j-detector-2021.12.20.jar 'c:/oraclexe/' >hits-20211221-oraclexe.txt

java -jar log4j-detector-2021.12.20.jar 'c:/Program Files/' >hits-20211221-programfiles.txt

java -jar log4j-detector-2021.12.20.jar 'c:/Program Files (x86)/' >hits-20211221-programfiles64.txt

java -jar log4j-detector-2021.12.20.jar 'c:/TEMP/' >hits-20211221-temp.txt

java -jar log4j-detector-2021.12.20.jar 'c:/Windows/' >hits-20211221-Windows.txt

java -jar log4j-detector-2021.12.20.jar 'c:/CAD Viewer 11/' >hits-20211221-CADViewer11.txt

java -jar log4j-detector-2021.12.20.jar 'c:/ProgramData/' >hits-20211221-programdata.txt

java -jar log4j-detector-2021.12.20.jar 'c:/ibm/' >hits-20211221-ibm.txt

java -jar log4j-detector-2021.12.20.jar 'c:/install/' >hits-20211221-install.txt