

Logging

CS 272 Software Development

Motivation

- Using a debugger is complicated in some settings
 - Multithreading, distributed computing
- Using `println` statements are not ideal
 - Easy to create, cumbersome to remove
- Using a custom Debug classes are not ideal
 - Easy to disable, but inefficient and inflexible

[Beginner's Guide to Quick Start Debugging](#) (in Eclipse)



Logging

- Next-level `println` debugging
 - Usually output to a log file instead of console
- Configurable without modifying code
 - Can configure **what** (errors, warnings, debug), **where** (console, file), and **how** (threads, line numbers, etc.)
- Usually have separate API and implementation



Logging Options in Java

- java.util.logging
<https://docs.oracle.com/en/java/javase/17/core/java-logging-overview.html>
- Simple Logging Facade for Java (SLF4J) • <http://www.slf4j.org/>
- LOGBack • <http://logback.qos.ch/>
- Apache Log4j2 • <https://logging.apache.org/log4j/2.x/>



Apache Log4j2

- A third-party library provided by Apache
- Supports multithreaded or distributed environments
- Efficient and flexible logging API and implementation
- Configurable without modifying source code
- Supports lambda expressions for log messages

<https://logging.apache.org/log4j/2.x/>



Simple Log4j2 Example

```
1. private static final Logger logger =  
2.     LogManager.getLogger();  
3.  
4. public static void main(String[] args) {  
5.     logger.info("Hello, World!");  
6. }
```

<https://logging.apache.org/log4j/2.x/manual/api.html>



Logging Levels

- **TRACE**

- Used for fine-grained debug messages
- Usually not shown or used unless really necessary

- **DEBUG**

- Used for normal debug messages
- Most commonly used level for logging

<https://logging.apache.org/log4j/2.x/log4j-api/apidocs/org/apache/logging/log4j/Level.html>



Logging Levels

- **INFO**

- Used for informational messages
- Often used for major events that were successful

- **WARN**

- Used when something concerning happened and there *might* be a problem

<https://logging.apache.org/log4j/2.x/log4j-api/apidocs/org/apache/logging/log4j/Level.html>



Logging Levels

- **ERROR**

- Used when an possibly recoverable error occurred
- Nearly always shown to the user

- **FATAL**

- Used when an irrecoverable error occurred
- Often used when about to exit prematurely

<https://logging.apache.org/log4j/2.x/log4j-api/apidocs/org/apache/logging/log4j/Level.html>



Logging Levels

- **ALL**

- Turns on all levels of logging, including TRACE
- Used to turn on all logging for debugging purposes

- **OFF**

- Turns off all levels of logging, including FATAL
- Used to disable logging, speeding up code

<https://logging.apache.org/log4j/2.x/log4j-api/apidocs/org/apache/logging/log4j/Level.html>



Logging Levels

- TRACE « lowest level, rarely used
- DEBUG « most common
- INFO « informational
- WARN « warnings
- ERROR « errors/exceptions
- FATAL « highest level, rarely used

<https://logging.apache.org/log4j/2.x/manual/architecture.html>



Configurable Output

- Configured via an XML, JSON, or YAML file
- Controls **what** information is output
- Controls **which classes** produce output
- Controls **which levels** are output
- Controls **where** messages are output

<https://logging.apache.org/log4j/2.x/manual/configuration.html>



Logging Configuration

- Appenders
 - Controls **where** log messages are output
 - Commonly the console and a log file
- Layouts
 - Control **what** information is output for an appender
 - Commonly the timestamp, level, and message

<https://logging.apache.org/log4j/2.x/manual/architecture.html>



Logging Configuration

- Loggers
 - Identified by a name (often class name)
 - Specify **level** of message to send to an appender
- Root Logger
 - Default logger, always accessible
 - Outputs ERROR messages to console by default

<https://logging.apache.org/log4j/2.x/manual/architecture.html>



Sample Configuration

```
1. <Configuration status="warn">
2.   <Appenders>
3.     <Console name="Console" target="SYSTEM_OUT">
4.       <PatternLayout pattern="%level: %message %n" />
5.     </Console>
6.   </Appenders>
7.   <Loggers>
8.     <Root level="error">
9.       <AppenderRef ref="Console" />
10.    </Root>
11.  </Loggers>
12. </Configuration>
```



Pattern Layout

- **%level** : level of the logging event
 - %level{length=1} to use a single letter
 - %level{lowerCase=true} to convert to lowercase
 - %level{WARN=Warning, ...} to change label
- **%message** : log message
- **%n** : platform dependent line separator (\n or \r\n)

<https://logging.apache.org/log4j/2.x/manual/layouts.html#PatternLayout>



Pattern Layout

- **%date{pattern}** : timestamp for logging event
 - %d{HH:mm:ss:SSS} to output just time
- **%thread** : thread name (useful later)
- **%location** : location where logging event created
 - Expensive operation, use with caution
 - See also %logger{}, %class{}, %method, and %line

<https://logging.apache.org/log4j/2.x/manual/layouts.html#PatternLayout>



Pattern Layout

Example Pattern

```
[%date{HH:mm:ss:SSS} %-5level{lowerCase=true}]  
%file@%line %t: %m%n
```

Example Output

```
[12:05:53:145 debug] CharacterCounter.java@181 main:  
Counting characters in file "src".
```

<https://logging.apache.org/log4j/2.x/manual/layouts.html#PatternLayout>



Apache Log4j2

- Popular and widely used logging tool
 - *Includes companies such as Apple, IBM, Oracle, Cisco, Google, Amazon, and more...*
- One of the top 50 most widely used open-source (non-npm) packages
- So widely used that...

<https://www.linuxfoundation.org/tools/census-ii-of-free-and-open-source-software-application-libraries/>
<https://www.cnn.com/2021/12/15/tech/log4j-vulnerability/index.html>



The Log4j security flaw could impact the entire internet. Here's what you should know

By Jennifer Korn

Updated 9:33 AM ET, Thu December 16, 2021

New York (CNN Business) — A critical flaw in widely used software has cybersecurity experts raising alarms and big companies racing to fix the issue.

The vulnerability, which was reported late last week, is in Java-based software known as "Log4j" that large organizations use to configure their applications -- and it poses potential risks for much of the internet.

Apple's cloud computing service, security firm Cloudflare, and one of the world's most popular video games, Minecraft, are among the many services that run Log4j, according to security researchers.

Jen Easterly, head of the Department of Homeland Security's Cybersecurity and Infrastructure Security Agency (CISA), called it "one of the most serious flaws" seen in her career. In a [statement](#) on Saturday, Easterly said "a growing set" of hackers are actively attempting to exploit the vulnerability.

As of Tuesday, more than 100 hacking attempts were occurring per minute, according to data this week from cybersecurity firm Check Point.

"It will take years to address this while attackers will be looking... on a daily basis [to exploit it]," said David Kennedy, CEO of cybersecurity firm TrustedSec. "This is a ticking time bomb for companies."

Here's what you should know:

What is Log4j and why does it matter?

Log4j is one of the most popular logging libraries used online, according to cybersecurity experts. Log4j gives software developers a way to build a record of activity to be used for a variety of purposes, such as troubleshooting, auditing and data tracking. Because it is both open-source and free, the library essentially touches every part of the internet.

"It's ubiquitous. Even if you're a developer who doesn't use Log4j directly, you might still be running the vulnerable code because one of the open source libraries you use depends on Log4j," Chris Eng, chief research officer at cybersecurity firm Veracode, told CNN Business. "This is the nature of software: It's turtles all the way down."

<https://www.cnn.com/2021/12/15/tech/log4j-vulnerability/index.html>

Nightmare before Christmas: What to know about the Log4j vulnerability

BY NICOLE SGANGA

UPDATED ON: DECEMBER 17, 2021 / 12:44 PM / CBS NEWS



<https://www.cbsnews.com/news/log4j-vulnerability-breach-patch/>



Upgrading Maven Version

←!— for logging →

```
<dependency>  
  <groupId>org.apache.logging.log4j</groupId>  
  <artifactId>log4j-api</artifactId>  
  <version>2.17.2</version>  
</dependency>
```

```
<dependency>  
  <groupId>org.apache.logging.log4j</groupId>  
  <artifactId>log4j-core</artifactId>  
  <version>2.17.2</version>  
</dependency>
```

```
<dependency>  
  <groupId>org.apache.logging.log4j</groupId>  
  <artifactId>log4j-slf4j-impl</artifactId>  
  <version>2.17.2</version>  
</dependency>
```

Always use latest version in `pom.xml` file!





CHANGE THE WORLD FROM HERE