

# PRODUCT SECURITY BULLETIN



Important information - Please read and keep

## Subject: Log4j Security Vulnerabilities (Log4Shell)

Date: 2024-02-12 (last update)

### Background

Security researchers at the Alibaba Cloud Security Team have discovered a new critical vulnerability in the Log4j java logging framework ([link](#)). Versions of Log4j affected include any version with version number  $\geq 2.0\text{-beta9}$  and  $< 2.16.0$ . The Log4j logging framework is very popular and often used by Java applications to facilitate logging. The vulnerability has been assigned the vulnerability identifier CVE-2021-44228 and has since been dubbed “Log4Shell” by security researchers.

Since Log4j is widely used and the vulnerability trivially exploitable (with exploits currently available on the internet), there are already reports in the media of exploitation of the vulnerability to attack systems.

In light of the Log4Shell vulnerability, security researchers have tested if previous versions of Log4j are vulnerable as well. While they found that Log4j 1.x does not contain the Log4Shell vulnerability, they discovered a separate vulnerability in Log4j 1.x. This vulnerability has been assigned the vulnerability identifier CVE-2021-4104.

### What do both vulnerabilities have in common?

CVE-2021-44228 and CVE-2021-4104 are different in terms of their attack vectors, although both vulnerabilities can be used to accomplish remote code execution (RCE) using special strings known as JNDI requests.

For both vulnerabilities, the JNDI request could contain a reference to an LDAP server, which the vulnerable Log4j versions would (unless specifically disabled by configuration) use to send a query to said LDAP server in order to substitute the text string with the response from the LDAP server. This mechanism can be exploited to fetch a specified Java class from a remote source and deserialize it, executing the class’s code in the process which compromises the system.

### What does Log4Shell (CVE-2021-44228) do?

For CVE-2021-44228, an attacker would try to get the system under attack to log JNDI requests, e.g. by inserting the request into the username field of a login form with the assumption that the username is logged for each authentication attempt. This could be carried out on local or remote systems running the affected Log4j versions.

Due to the trivial nature of the exploit, the possible consequences and the remote attack vector, CVE-2021-44228 has received the highest possible CVSS 3.1 (Common Vulnerability Scoring System) base score of **10.0 (Critical)**.

# PRODUCT SECURITY BULLETIN



Important information - Please read and keep

What does CVE-2021-4101 do?

The flaw behind CVE-2021-4104 is in the use of JMSAppender within the Apache Log4j library. Applications running version 1.2.x with JMSAppender enabled and with JMSAppender set to allow JNDI requests run the risk of an attacker performing an exploit that can lead to remote code execution. In a different scenario, an attacker with write access to the Log4j configuration could set the JMSAppender to their own JMS Broker. This would allow the attacker to send JNDI requests to the application.

CVE-2021-4104 received a CVSS 3.1 base score to **6.6 (Medium)** due to the local attack vector and JMSAppender not being enabled by default.

## Response

Our Affected Products

Log4Shell (CVE-2021-44228)

Our products **do not make use of the Log4j library versions** that is vulnerable to the Log4Shell vulnerability and are therefore **not vulnerable** to any exploits.

CVE-2021-4101

Our Respiratory Diagnostics' SentryConnect solution, which is utilized to connect SentrySuite™ solutions to hospital information systems, utilizes [Mirth Connect](#) for integration purposes. Mirth Connect utilizes an [older version of Log4j](#) that is **potentially vulnerable to CVE-2021-4104**, but **not vulnerable to CVE-2021-44228**. Mirth Connect does not make use of the vulnerable JMSAppender in its Log4j configuration by default.

How are we responding?

We have implemented the changes below around the disclosure of the Log4j vulnerabilities from **SentryConnect Gateway version 6.0.2.1** onwards.

- Removed legacy properties from mirth.properties file regarding encryption, digest and security
- Added random strong passwords for keystore generation
- Removed deprecated https ciphersuites
- Updated MirthConnectAdminLauncher to version 1.3.1
- TLSv1.3 Support added
- TLSv1.1 Support removed
- Upgraded to MirthConnect 4.2.0

For Global Distribution.

© 2025. Jaeger Medical GmbH. All rights reserved. Jaeger, the Jaeger Medical logo and all other trademarks or registered trademarks are property of Jaeger Medical GmbH or one of its affiliates. VYR-GBL-2500069| 1.0

 Jaeger Medical GmbH  
Leibnizstrasse 7  
97204 Hoechberg  
Germany

 0123

# PRODUCT SECURITY BULLETIN



Important information - Please read and keep

- Upgraded to openJDK 17.0.5.8 (LTS)
- Upgraded MirthConnect's jetty 9.4.44 to jetty 9.4.49

## Mitigations & Compensating Controls

### Log4Shell (CVE-2021-44228)

Our products are not affected by Log4Shell, therefore no mitigations for Log4Shell are required.

### CVE-2021-4101

We recommend applying the following mitigations and compensating controls:

- Customers using versions before SentryConnect Gateway version 6.0.2.1 should reach out to [support.connect.eu@jaegerrdx.com](mailto:support.connect.eu@jaegerrdx.com) to initiate the update process.
- Until the update process is complete, please review the access controls to ensure only authorized personnel has write-level access to Log4j configuration files.

For product or site-specific concerns, contact your Jaeger Medical service representative.

For more information on the Jaeger Medical proactive approach to product security and vulnerability management, contact us at [productsecurity@jaegerrdx.com](mailto:productsecurity@jaegerrdx.com) or visit [www.jaegerrdx.com/product-security](http://www.jaegerrdx.com/product-security).