December 12, 2021

Last Update – December 15, 2021

# Oracle E-Business Suite Security Analysis

## Log4j Vulnerabilities Impact on Oracle E-Business Suite CVE-2021-44228, CVE-2021-45046, and CVE-2021-4104

## Background

Apache Log4j is a commonly used logging library for Java applications. A critical risk and two medium risk security vulnerabilities have been discovered in the Log4j library. The first vulnerability, CVE-2021-44228 and nicknamed Log4Shell, is a critical risk vulnerability that allows an attack to potentially perform remote code execution on the application server. Log4j introduced a feature in version 2.0 of the library that allows for log entries to lookup information through the Java Naming and Directory Interface (JNDI). If the application is configured to use Log4j and writes log entries with end-user input, an attacker is able to invoke JNDI calls which in turn allow for download and execution of arbitrary Java classes. This vulnerability is especially problematic in that an attacker is able to spray attacks across the Internet and vulnerable servers will respond to the attacker's server.

The second vulnerability, CVE-2021-45046, is a variation of the first vulnerability and allows bypass of the protections for CVE-2021-44228. The Log4j configuration setting "formatMsgNoLookups" used to mitigate the risk for CVE-2021-44228 is not effective for this variation of the vulnerability as the Thread Context Map (MDC) pattern is exploited rather than the message pattern (%m).

A third vulnerability, CVE-2021-4104, has been discovered in unsupported versions of Log4j (1.x) that allow for Java deserialization attacks when the Log4j JMSAppender is configured. The JMSAppender publishes log entries or events to Java Message Service (JMS) topics.

## Oracle E-Business Suite (EBS) and Log4j

The Log4j library is delivered with Oracle EBS versions 12.0 through 12.2.10. Depending on the Oracle EBS version and installed Oracle EBS patches, between 5 and 30 instances of the Log4j library will installed on each Oracle EBS application, concurrent manager, and database server. The installed Log4j versions will include both 1.x and 2.x. You can identify the instances of the log4j library and configuration file on a specific server by running the Linux command "find / -name log4j*". The vulnerable libraries are "log4j-core.jar" and "log4j_x_y_x.jar" such as "log4j_1_2_13.jar". "log4j-core.jar" is version 2.x of the library potentially vulnerable to CVE-2021-44228 and CVE-2021-45046 and "log4j_x_y_x.jar" is version 1.x of the library and potentially vulnerable to CVE-2021-4104.

In order for the vulnerability to be exploitable, three conditions must be met –

1. The Log4j library is defined in the Java classpath. For the Oracle EBS web application, the primary concern is the web application Java containers including oacore, forms, oafm, and forms-c4ws and for 12.2 the WebLogic console.
2. Log4j must be configured to log events, which can be simply done by initiating a logger in the code even without a Log4j configuration file.
3. The Log4j logging api be called (e.g., logger.info) and un-filtered end-user input is passed to the logger. Most of the vulnerable applications on the Internet use Log4j to log all HTTP requests or similar activity thus logging user-manipulatable HTTP headers such as user-agent and full URLs.

For the Oracle EBS Java containers (oacore, forms, oafm, and forms-c4ws) and WebLogic console, the following table shows the versions and locations of where the Log4j libraries are loaded –

| Oracle EBS Version | Log4j Versions |
|---|---|
| **12.2** | consoleapp = 1.2.8<br>oacore = 1.2.13 and 2.11.1[x]<br>forms = no<br>oafm = 1.2.13 and 2.11.1[x]<br>forms-c4ws = no<br><br>[x] 2.11.1 installed with when R12.TXK.C.DELTA.12+ is applied |
| **12.1** | none |
| **12.0** | none |
| **11.5** | none |

To identify which Log4j versions are installed, you can run the following query and you are looking for either “log4j_core.jar” (2.x) or “log4j_x_y_z.jar” (1.x).

```
SELECT *
FROM applsys.ad_files
WHERE filename like 'log4j%.jar';
```

## Oracle E-Business Suite Impact

Vulnerable versions of the Log4j library are loaded in some of the Oracle EBS web application Java containers for all Oracle EBS 12.2 environments. If R12.TXK.C.DELTA.12 or later is applied, then version 2.11.1 is installed and prior to R12.TXK.C.DELTA.12 then 1.2.13 will be installed. For the WebLogic console application, Log4j 1.2.8 will be installed.

**Integrigy and Oracle Corporation have not been able to identify any locations in the standard Oracle EBS web application components running in the Oracle EBS Java containers nor in the WebLogic console where a Log4j logger would be initiated, or any log entries would be written using Log4j.** Therefore, an exploitable attack vector has not yet been discovered within Oracle EBS. This is applicable for all versions of Oracle EBS 12.2 regardless if Log4j 1.x or 2.x is used.

The primary risk and concern are that a customer customization or third-party Oracle EBS add-on or integration may be using the Log4j library, therefore, introduce an attack vector to allow one of these Log4j vulnerabilities to be exploitable. For Log4j 2.x, this would require end-user input to be logged using Log4j and for Log4j 1.x would require the JMSAppender to be configured in the Log4j configuration. In our research, Integrigy has identified a client customization where Log4j is actively used and a third-party product that leverages Log4j 1.x but does not enable the JMSAppender.

**With the release of CVE-2021-45046, the Oracle initial work-around to implement the LOG4J_FORMAT_MSG_NO_LOOKUPS environment variable was not a complete fix.** Oracle has updated the guidance in My Oracle Support Note ID 2827804.1 to remove the vulnerable class from the JAR file. Per the Log4j website, "Other insufficient mitigation measures are: setting system property log4j2.formatMsgNoLookups or environment variable LOG4J_FORMAT_MSG_NO_LOOKUPS to true for releases >= 2.10, or modifying the logging configuration to disable message lookups with %m{nolookups}, %msg{nolookups} or %message{nolookups} for releases >= 2.7 and <= 2.14.1. The safest thing to do is to upgrade Log4j to a safe version, or remove the JndiLookup class from the log4j-core jar."

In order to completely mitigate these vulnerabilities in an Oracle EBS 12.2 environment, Integrigy recommends the following actions –

1. If R12.TXK.C.DELTA.12 or later is applied (see above to determine), the recommended approach based on the new vulnerability (CVE-2021-45046) is to remove the vulnerable JNDILookup class from the Log4j JAR file. Use mitigation procedure in My Oracle Support Note ID 2827804.1.

2. Review all Oracle EBS web customizations to verify Log4j is not used in any customizations. As how Log4j will be used is highly dependent on the customization, review code for any usage of log4j such as Java imports in code like "import org.apache.logging.log4j.Logger;". Log4j also may be used through SLF4J (Simple Logging Façade for Java) where Log4j is not in the code but will require the Java library "log4j-slf4j-impl.jar" and a Log4j configuration file may exist under "resources".

3. Review all Oracle EBS third-party products that integrate with the Oracle EBS web application. Typically, these third-party products will require an AutoConfig customization to one of the "*web_xml_FMW.tmp" files and these files will be located in $FND_TOP/admin/template/custom. To load libraries, these JAR files may be copied to different locations such as $RUN_BASE/EBSapps/comn/shared-libs/ebs-3rdparty/WEB-INF/lib. Review the product installation documentation and code to determine if log4j_core.jar or log4j-slf4j-impl.jar are included or installed. Contact vendor technical support to verify if Log4j is used in the product.

To address to the potential risk of vulnerable customizations or third-party products, Integrigy recommends AppDefend be implemented in your Oracle EBS environment to prevent exploitation of these vulnerabilities.

All Oracle EBS ancillary components should be reviewed to determine if these products may be vulnerable. This includes all identity management, reporting, backup, and management applications. As Java is extensively used for development of Oracle products and a number of products typically implemented with Oracle EBS were acquired by Oracle, the use of vulnerable Log4j versions within these ancillary components is a distinct risk. As of this last update, the following EBS ancillary products are vulnerable and must be patched –

- Oracle E-Business Suite with Enterprise Command Centers
- Oracle E-Business Suite with Oracle E-Business Suite Extensions for Endeca
- Oracle Agile
- Oracle Hyperion
- Oracle Enterprise Manager

## AppDefend and Log4j

AppDefend was updated to address the Log4j "Log4Shell" vulnerability. To protect Oracle E-Business Suite, AppDefend has been updated to address these Log4Shell vulnerability two different ways –

1. AppDefend injects the fix through the system parameter into the Java containers blocking JNDI without needing to apply the Oracle fix or future patch
2. New AppDefend rules that detect and block Log4j attack strings. These rules block most variations of the attack strings.

## AppDefend and AppSentry Not Vulnerable

Integrigy AppDefend and AppSentry do not use the Log4j library for logging and are not vulnerable to these security bugs.  Both products use another Java logging library that does not have the JNDI feature.

## References

- Apache Log4j Security Alert CVE-2021-44228 Products and Versions (Doc ID 2827611.1)
- CVE-2021-44228 Advisory for Oracle E-Business Suite (Apache log4j Vulnerabilities) (Doc ID 2827804.1)
- Apache Log4j Security Vulnerabilities, https://logging.apache.org/log4j/2.x/security.html

## History

December 10, 2021 – Initial Internal Analysis
December 12, 2021 – Internal Draft
December 13, 2021 – AppDefend Updated
December 13, 2021 – Added CVE-2021-4104
December 14, 2021 – First Version Published
December 15, 2021 – Added CVE-2021-45046

## About Integrigy

Integrigy Corporation is a leader in application security for enterprise mission-critical applications and databases. AppSentry, our ERP application and database security assessment tool, assists companies in securing their largest and most important applications and databases through detailed security audits and actionable recommendations. AppDefend, our enterprise web application firewall is specifically designed for the Oracle E-Business Suite and PeopleSoft. Integrigy Consulting offers comprehensive security assessment services for leading databases and ERP applications, enabling companies to leverage our in-depth knowledge of this significant threat to business operations.

Integrigy Corporation
14 Westlake Drive
Nashville, Tennessee 37205 USA
888/542-4802
www.integrigy.com



If you have any questions, comments or suggestions regarding this document, please send them via e-mail to info@integrigy.com.

The Information contained in this document includes information derived from various third parties. While the Information contained in this document has been presented with all due care, Integrigy Corporation does not warrant or represent that the Information is free from errors or omission. The Information is made available on the understanding that Integrigy Corporation and its employees and agents shall have no liability (including liability by reason of negligence) to the users for any loss, damage, cost or expense incurred or arising by reason of any person using or relying on the information and whether caused by reason of any error, negligent act, omission or misrepresentation in the Information or otherwise.

Furthermore, while the Information is considered to be true and correct at the date of publication, changes in circumstances after the time of publication may impact on the accuracy of the Information. The Information may change without notice.

Integrigy's Vulnerability Disclosure Policy – Integrigy adheres to a strict disclosure policy for security vulnerabilities in order to protect our clients. We do not release detailed information regarding individual vulnerabilities and only provide information regarding vulnerabilities that is publicly available or readily discernible. We do not publish or distribute any type of exploit code. We provide verification or testing instructions for specific vulnerabilities only if the instructions do not disclose the exact vulnerability or if the information is publicly available.