Security Bulletin

Issue

New information has been published by Nevis Security regarding the exploit in log4j that allows remote code execution (CVE-2021-44228, CVE-2021-4104 and CVE-2021-45046). This security bulletin informs you about the new risk for the Nevis components. nevisProxy itself is not at risk (no Java component) but can help to protect the applications behind it.

We also append this time some suggestions from Nevis Security regarding implementation of ModSecurity filters to protect applications behind nevisProxy. Those suggestions are provided as-is without a guarantee that the applications are 100% protected and without the guarantee that nothing will break. They can only be used as a temporary measure until the applications are protected properly.

Date

Zurich, 15.12.2021

Published by

Adnovum Informatik AG
Badenerstrasse 170
8004 Zurich
Switzerland
+41 44 272 6111
info@adnovum.ch, www.adnovum.ch

Description

CVE-2021-44228 and CVE-2021-4104

The widely used Java logging library log4j has an unauthenticated RCE vulnerability (CVE-2021-44228 and CVE-2021-4104), which allows an attacker to take control of the affected server if user-controlled strings are logged. This security bulletin focuses on the risk that exists with all versions of log4j regarding the use of JMSAppender. Although rather unlikely, we advise all our clients as a precautionary measure to make sure they have not configured a JMSAppender in their log4j configurations in the list of components below. If you do have a JMSAppender enabled for one of those components, please evaluate the Recommendations / Measures paragraph below and, if necessary, get in touch with your support contacts at Adnovum. Even if you have configured a JMSAppender, the exploitation difficulty of this is significantly higher than the first attack vector described in our previous security bulletin (2021-06) and the Nevis security bulletin. It is not remotely or directly exploitable but only as a

secondary attack vector if all unlikely conditions are met (see Risk Assessment and Recommendations below).

CVE-2021-45046

CVE-2021-45046 has been opened because it was found that the immediate fix for CVE-2021-44228 (setting log4j2.noFormatMsgLookup to "true") does not work in all cases. This would impact the two Nevis components (nevisIDM and nevisReports) and the versions mentioned in our last security bulletin. Nevis Security is still analyzing the exact impact. Although so far it is clear that it does not affect those components in their default configurations, customers should make sure they do no use any of the new sensitive patterns in case they have configured their own log4j log format/layout.

New sensitive patterns:

Context Lookup (for example, $${ctx:loginId}) or a Thread Context Map pattern (%X, %mdc, or %MDC).

Please note that this is a preliminary information, before the analysis is completed. We decided to add it to this security bulletin anyway.

Affected Components (CVE-2021-44228 and CVE-2021-4104)

Component	Versions	log4j configuration file
nevisIDM	LTS19	/var/opt/nevisidm/<instance>/conf/log4j.xml
nevisAuth	all versions	/var/opt/nevisauth/<instance>/conf/log4j.xml
nevisLogRend	all versions	/var/opt/nevislogrend/<instance>/conf/log4j.xml
nevisAgent	all versions	/var/opt/nevisagent/default/conf/log4j.properties
nevisAdmin 3	all versions	/var/opt/adnglassfish/nevisadmin/conf/log4j.xml
nevisMeta	all versions	/var/opt/nevismeta/<instance>/conf/log4j.xml
nevisFIDO	all versions	/var/opt/nevisfido/<instance>/conf/log4j.xml
nevisDataporter	all versions	/var/opt/nevisdp/<instance>/conf/log4j.xml
ninja	all versions	

Risk Assessment

The risk is rated medium, as it requires special conditions to be affected.

You are affected if all of the following conditions are true:

- You are using a component from the list of affected components.
- You have a JMSAppender configured in the log4j configuration of that component.

- You have configured a JNDI type lookup in any of the two parameters mentioned in Recommendations / Measures.

In this case, make sure to break any or both of the last two conditions above.

Recommendations / Measures

If you have configured a JMSAppender, it is important that you have not configured a JNDI type lookup (for example, “ldap::://internal-host.com”) as part of the following two parameters:

- *TopicBindingName*, or
- *TopicConnectionFactoryBindingName*

Even if a JNDI type of value is configured, exploitation is still a secondary attack vector that requires the exploitation of the internal system those properties point to.

If JNDI type lookup is not configured, exploitation requires an internal actor (administrator) setting this value to a third-party or internally compromised system. If this vector is of concern to you, we recommend against using a JMSAppender for the affected components.

In order to check if a JMSAppender is configured, you have to inspect your log4j configuration file. If you find a line containing the org.apache.log4j.net.JMSAppender, you may be vulnerable.

As an advanced use case, it may be possible to also configure this with custom classes. If you are using custom classes in your components, verify if you have configured a JMSAppender. One indication of custom class configuration is that the log4j.configuratorClass system property is passed to the JVM.